

MMUSIC  
Internet-Draft  
Expires: March 9, 2008

R. Walsh  
J-P. Luoma  
Nokia  
J. Peltotalo  
S. Peltotalo  
Tampere University of Technology  
J. Greifenberg  
Universitaet Bremen  
September 6, 2007

**The IMG Envelope**  
**draft-walsh-mmusic-img-envelope-07**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on March 9, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).

## Abstract

This document defines the metadata transfer envelope for Internet Media Guides (IMGs). IMG metadata describes files, resources and multimedia programs available for streaming or downloading via multicast or unicast. IMG metadata is encapsulated into, or associated with, an IMG envelope before actual transport. The IMG envelope is a structure providing independence between IMG transport protocols and different metadata formats. This specification provides the IMG envelope instantiation using structured Extensible Markup Language (XML) syntax, both as a wrapper in which to embed an IMG metadata object and as a distinct object to associate with a distinct IMG metadata object.

## Table of Contents

<a href="#">1.</a>	<a href="#">Introduction . . . . .</a>	<a href="#">4</a>
<a href="#">2.</a>	<a href="#">Conventions Used in This Document . . . . .</a>	<a href="#">5</a>
<a href="#">3.</a>	<a href="#">IMG Envelope Usage . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.</a>	<a href="#">Applicability of an IMG Envelope . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.1.</a>	<a href="#">The Two IMG Envelope Cases . . . . .</a>	<a href="#">6</a>
<a href="#">3.1.2.</a>	<a href="#">Relationship with IMG Data Types . . . . .</a>	<a href="#">7</a>
<a href="#">3.1.3.</a>	<a href="#">Relationship with IMG Operations . . . . .</a>	<a href="#">7</a>
<a href="#">3.2.</a>	<a href="#">IMG Envelope Characteristics . . . . .</a>	<a href="#">7</a>
<a href="#">3.2.1.</a>	<a href="#">Versioning of the IMG Metadata Fragment . . . . .</a>	<a href="#">7</a>
<a href="#">3.2.2.</a>	<a href="#">Detecting the IMG Metadata Fragment Format Type . . . . .</a>	<a href="#">8</a>
<a href="#">3.2.3.</a>	<a href="#">Securing IMG Envelope Integrity . . . . .</a>	<a href="#">8</a>
<a href="#">3.2.4.</a>	<a href="#">Reliable Delivery of IMG Envelope and IMG Metadata Fragment . . . . .</a>	<a href="#">8</a>
<a href="#">3.2.5.</a>	<a href="#">Consistency Checking for IMG Metadata . . . . .</a>	<a href="#">9</a>
<a href="#">3.2.6.</a>	<a href="#">Administrative Scope . . . . .</a>	<a href="#">9</a>
<a href="#">3.2.7.</a>	<a href="#">Resolving IMG Metadata Fragments to Resource Locations . . . . .</a>	<a href="#">9</a>
<a href="#">3.2.8.</a>	<a href="#">Describing Multiple IMG Metadata Fragments within an Index Referencing Envelope . . . . .</a>	<a href="#">10</a>
<a href="#">3.2.9.</a>	<a href="#">Using Multipart MIME to bind IMG Metadata Fragment Delivery . . . . .</a>	<a href="#">10</a>
<a href="#">4.</a>	<a href="#">IMG Envelope Format . . . . .</a>	<a href="#">11</a>
<a href="#">4.1.</a>	<a href="#">IMG Envelope Semantics . . . . .</a>	<a href="#">11</a>
<a href="#">4.2.</a>	<a href="#">XML Syntax for the IMG Envelope . . . . .</a>	<a href="#">12</a>
<a href="#">5.</a>	<a href="#">Security Considerations . . . . .</a>	<a href="#">15</a>
<a href="#">6.</a>	<a href="#">IANA Considerations . . . . .</a>	<a href="#">16</a>
<a href="#">6.1.</a>	<a href="#">Registration Request for XML Schema of IMG Envelope . . . . .</a>	<a href="#">16</a>
<a href="#">6.2.</a>	<a href="#">Considerations for application/envelope+xml Media-Type Registration . . . . .</a>	<a href="#">16</a>
<a href="#">6.2.1.</a>	<a href="#">Media-Type Registration Request for application/envelope+xml . . . . .</a>	<a href="#">16</a>



<a href="#">7.</a>	<a href="#">Acknowledgements</a>	<a href="#">18</a>
<a href="#">8.</a>	<a href="#">References</a>	<a href="#">19</a>
<a href="#">8.1.</a>	<a href="#">Normative References</a>	<a href="#">19</a>
<a href="#">8.2.</a>	<a href="#">Informative References</a>	<a href="#">19</a>
<a href="#">Appendix A.</a>	<a href="#">IMG Envelope Examples</a>	<a href="#">21</a>
<a href="#">A.1.</a>	<a href="#">SDP Metadata Fragment Embedded in an IMG Envelope</a>	<a href="#">21</a>
<a href="#">A.2.</a>	<a href="#">An IMG Envelope Referencing an XML Metadata Fragment</a>	<a href="#">21</a>
<a href="#">A.3.</a>	<a href="#">An Index Referencing Envelope</a>	<a href="#">22</a>
<a href="#">Appendix B.</a>	<a href="#">IMG Architecture Terminology</a>	<a href="#">23</a>
<a href="#">Appendix C.</a>	<a href="#">IMG Architectural Description</a>	<a href="#">27</a>
<a href="#">C.1.</a>	<a href="#">IMG Metadata Structure and Fragmentation</a>	<a href="#">27</a>
<a href="#">C.1.1.</a>	<a href="#">Fragmentation</a>	<a href="#">27</a>
<a href="#">C.1.2.</a>	<a href="#">Fragments within Fragments</a>	<a href="#">27</a>
<a href="#">C.2.</a>	<a href="#">IMG Metadata Discovery</a>	<a href="#">27</a>
<a href="#">C.2.1.</a>	<a href="#">Initial IMG Metadata Discovery</a>	<a href="#">27</a>
<a href="#">C.2.2.</a>	<a href="#">IMG Metadata Update Discovery</a>	<a href="#">28</a>
C.3.	<a href="#">Support for Non-contiguous IMG Metadata Fragment</a>	
Version Series		<a href="#">30</a>
<a href="#">C.4.</a>	<a href="#">Detecting the IMG Metadata Fragment Format Type</a>	<a href="#">30</a>
<a href="#">C.5.</a>	<a href="#">Reliable Delivery of IMG Metadata</a>	<a href="#">31</a>
C.6.	<a href="#">Parsing and Storage of IMG Metadata Related to Delta</a>	
Types		<a href="#">31</a>
<a href="#">C.7.</a>	<a href="#">Security Considerations</a>	<a href="#">31</a>
Authors' Addresses		<a href="#">33</a>
Intellectual Property and Copyright Statements		<a href="#">34</a>



## **1. Introduction**

This document defines the format and use of Internet Media Guide (IMG) envelope. The scope and background of the work on Internet Media Guides have been described in the IMG requirements [\[2\]](#) and IMG framework [\[3\]](#) specifications.

The purpose of the IMG metadata is to provide machine and human readable information describing files, resources and multimedia programs available for streaming or downloading via multicast or unicast. IMG metadata is encapsulated into, or associated with, an IMG envelope before it is passed to an IMG transport protocol. The purpose of the IMG envelope is to provide independence of metadata formats from transport protocols, and to enable versioning, updating and expiring of transmitted metadata. This specification provides the IMG envelope instantiation using structured Extensible Markup Language (XML) syntax [\[17\]](#), both as a wrapper in which to embed an IMG metadata object and as a distinct object to associate with a distinct IMG metadata object.



## **2. Conventions Used in This Document**

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC 2119](#) [[1](#)].

IMG specific definitions can be found in the IMG architecture terminology (Appendix B).



### 3. IMG Envelope Usage

#### 3.1. Applicability of an IMG Envelope

A single IMG envelope shall describe a single IMG metadata fragment, and thus instances of the two are paired.

##### 3.1.1. The Two IMG Envelope Cases

An instance of IMG envelope shall be associated with an instance of IMG metadata fragment by one of two methods:

- Embedded: The IMG metadata fragment is embedded within the IMG envelope.
- Referenced: The IMG metadata fragment is referenced from the IMG envelope.

Figures 1a and 1b illustrate the embedded and referenced cases.

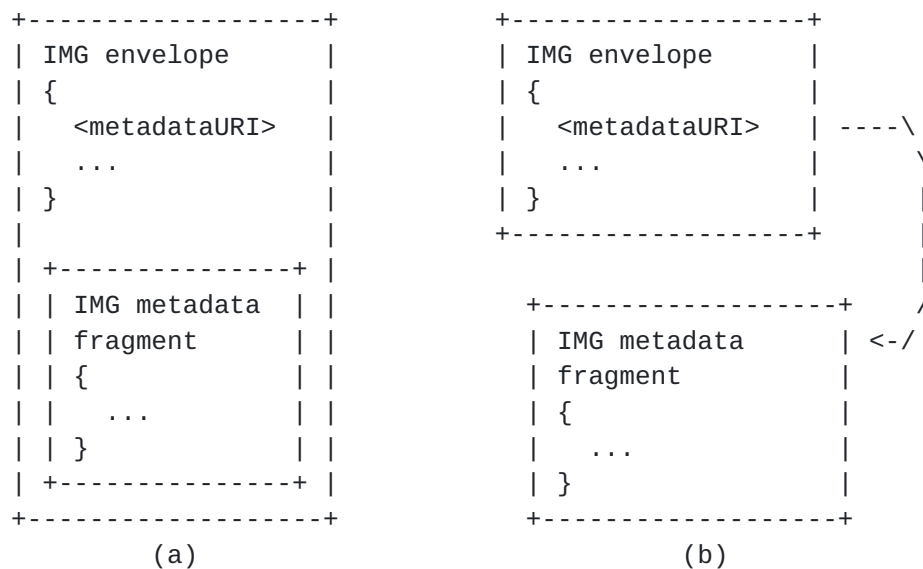


Figure 1: IMG Envelope: (a) Embedded Case, (b) Referenced Case

In the embedded case, the IMG envelope and the IMG metadata fragment are, by definition, transported together and in-band of one another. In the referenced case, the IMG envelope and the IMG metadata fragment MAY be transported together and in-band, or out-of-band of one another using different transport sessions.

An IMG sender SHALL make an IMG envelope instance available for each IMG metadata fragment instance. The creation and use of both an



embedded IMG envelope instance and a referenced IMG envelope instance for a particular IMG metadata fragment instance is permitted, but is not generally expected.

Detailed discussion of the transport of IMG envelope and IMG metadata fragment are beyond the scope of this document, however, it is anticipated that IMG envelopes will be transported at least as often than their respective IMG metadata fragments.

### **[3.1.2.](#) Relationship with IMG Data Types**

An instance of an IMG envelope describes a certain instance of a certain IMG metadata fragment, irrespective of whether the complete, delta or pointer data type functionality is used. The content type (MIME type) of the IMG metadata fragment is used to differentiate complete and delta descriptions of an IMG metadata fragment, as described in [Section 3.2.2](#).

A referencing IMG envelope provides the pointer data type functionality by itself and SHOULD be used for this purpose where ever the pointer data type is implemented.

### **[3.1.3.](#) Relationship with IMG Operations**

An IMG envelope is used with the following logical IMG operations: IMG ANNOUNCE, IMG NOTIFY and IMG RESOLVE.

## **[3.2.](#) IMG Envelope Characteristics**

### **[3.2.1.](#) Versioning of the IMG Metadata Fragment**

The version of an IMG metadata fragment SHALL be identified by a version field in the IMG envelope, irrespective of the IMG data type (i.e. for complete, delta and pointer types alike). The metadataURI ("metadataURI" is defined in [Section 4](#) of this document) and version number SHALL resolve to a unique instance of the IMG metadata fragment (and its paired IMG envelope). The level of this uniqueness is dependent on the administrative scope of the metadataURI namespace and the version control.

Note: The same version number is inherited to the IMG envelope as IMG envelope and IMG metadata fragment instances occur in matched pairs. Thus, there is no need for an additional "version number of the IMG envelope" attribute.



### **3.2.2. Detecting the IMG Metadata Fragment Format Type**

The IMG envelope SHALL provide a content type field. This field MUST provide the MIME type of the IMG metadata fragment when the IMG metadata fragment is embedded in the IMG envelope. This field MAY be used when the IMG metadata fragment is not embedded.

### **3.2.3. Securing IMG Envelope Integrity**

In general, the IMG envelope data SHOULD NOT be encrypted, although it can be signed. Unencrypted IMG envelope data allows IMG transceivers to cache and maintain IMG metadata without being required to be a trusted party able to decrypt the secure data.

Note, an IMG system aside from the public Internet may chose to trust IMG transceivers, or exclude transceivers entirely. In these cases, and where no bearer-specific security method is used, there may be compelling reasons to encrypt this IMG envelope data and, since in this context the encryption of IMG envelope data presents no additional limitations, the previous recommendation may be ignored.

IMG envelopes exposed to non-secure connections on the public Internet SHOULD be signed to lessen the risk of security attacks associated with delivery.

The signature, and possible encryption, method(s) used are very much IMG envelope syntax and application specific. For example, one could use S/MIME [15] as the content encoding type for IMG metadata objects with an authentication wrapper, and one could use XML-DSIG [16] to digitally sign an IMG metadata fragment or an IMG envelope. Further specification on securing IMG envelopes is beyond the scope of this document.

### **3.2.4. Reliable Delivery of IMG Envelope and IMG Metadata Fragment**

The importance of the IMG envelope data and its timely delivery, relative to its associated IMG metadata fragment, will vary from one application and deployment to another. Where knowledge of data consistency (envelope usage) is a higher priority than ensuring perfect data consistency (metadata fragment usage) then it would be prudent to ensure the same or higher levels of reliability for the IMG envelope data, and vice versa.

Providing similar levels of reliable delivery overhead for both the IMG envelope and IMG metadata fragment is a balanced approach.

This would imply the same reliability method for the IMG envelope and the IMG metadata fragment pair. However, it does not imply the same



level of reliability as, in the case that a discrete IMG envelope object is significantly smaller in size than its discrete IMG metadata object, there will be a greater loss multiplier effect for the larger object (as it would be delivered using more IP packets providing a higher probability that one or more is lost in transit).

Note: Providing a similar level of reliability overhead for an embedded IMG metadata fragment as its embedding IMG envelope is trivial since they are transported as a single object.

#### **3.2.5. Consistency Checking for IMG Metadata**

The IMG envelope SHALL support time stamps to set the start and end times of the IMG metadata fragment applicability. The end time sets the expiry time for the IMG metadata fragment, so that: (a) a receiver would know that it needs to check whether its metadata is consistent with the sender, (b) if the IMG metadata fragment is no longer of use it may be discarded, (c) the same IMG metadata fragment version may be unchanged but have its time validity changed (so the client would know that an update of the IMG metadata fragment is unnecessary although the expiry is extended or shortened). The start time may be used to postpone the use of an IMG metadata fragment until some future time.

#### **3.2.6. Administrative Scope**

The definition of any administrative scope for source, aggregation or proxy functions on IMG metadata and IMG envelope is out of the scope of this document. It is assumed that the same administrative domain applies to both IMG envelope and IMG metadata fragment of a specific pair (note, this does not imply that they originate from the same source or even same domain). It is also assumed that the namespace is consistent with each name (URI) identifying only a single resource (although, naturally it may identify multiple instances/versions of that resource).

Where the administrative domain does not impose its own naming conventions on the IMG envelope, the following naming convention SHOULD be used for the IMG envelope name (URI):

envelopeURI = metadataURI + ".env"

Note: "metadataURI" is defined in [Section 4](#) of this document.

#### **3.2.7. Resolving IMG Metadata Fragments to Resource Locations**

The IMG envelope MAY provide the mechanism to identify multiple locations which a single IMG metadata fragment may be found.





"alternativeURL", defined in [Section 4](#), is used for this purpose.

#### **[3.2.8](#). Describing Multiple IMG Metadata Fragments within an Index Referencing Envelope**

It may be more efficient to describe multiple IMG metadata fragments in a single envelope object. Therefore an index referencing envelope may describe multiple IMG metadata fragments as different items within the envelope object, i.e., multiple referencing IMG envelopes are encapsulated into a single envelope object.

#### **[3.2.9](#). Using Multipart MIME to bind IMG Metadata Fragment Delivery**

In order to bind several IMG metadata fragments into a single transport object Multipart MIME may be used. In this case the first object of Multipart MIME should be an index referencing envelope as an index to all IMG metadata fragments.



## **4. IMG Envelope Format**

An IMG metadata fragment SHALL be encapsulated into or associated with an IMG envelope before it is passed to an IMG transport protocol for delivery. The IMG envelope enables each IMG metadata fragment to be uniquely identified and versioned in a uniform way independent of the particular IMG transport protocol used for delivery. The same IMG envelope format is used for the logical operations IMG ANNOUNCE, IMG NOTIFY and IMG RESOLVE.

The next section describes the mandatory semantics for any IMG envelope format. The section after that describes the XML instantiation of the IMG envelope. For maximum interoperability, the given XML syntax SHOULD be used for textual representation of the IMG envelope. However, an IMG transport protocol MAY specify the use of an additional IMG envelope syntax, possibly providing a binary encoding of the XML format of this document.

### **4.1. IMG Envelope Semantics**

The following fields can be associated with the IMG metadata fragment. The fields are mandatory to include unless marked as optional.

The following fields are described in the IMG envelope and thus associated with the respective IMG metadata fragment. Each field SHALL be included in any IMG envelope, except where specifically marked as optional.

- o metadataURI: A URI providing a unique identifier for the IMG metadata fragment.
- o alternativeURL: An optional one or multiple alternative URLs to locate metadata resource by different schemes. Where the metadataURI is URL it provides the first of the list of URLs.
- o version: The version number of the associated instance of the IMG metadata fragment. The version number should be initialized to one. The version number shall be increased by one whenever the IMG metadata fragment is updated.
- o validFrom: The date and time from which the IMG metadata fragment is valid. (Optional). If not used, the receiver SHOULD assume the IMG metadata fragment version is effective immediately.
- o validUntil: The date and time when the IMG metadata fragment expires. This sets the expiry time for the IMG metadata fragment.



(Optional).

- o contentType: The MIME type of the IMG metadata fragment. For textual representation, this MUST be used as defined for "Content-Type" in [4]. For IMG envelopes which embed their IMG metadata fragment this attribute is mandatory. For associations by reference (not embedded) this field is optional.

An IMG envelope instantiation syntax MUST provide clear rules on the determination of embedded IMG metadata fragment start and end boundaries. Rules of how to avoid confusing the IMG envelope parser with data in the IMG metadata fragment (e.g. which resembles envelope data format) MUST be provided with any IMG envelope syntax specification.

#### **4.2. XML Syntax for the IMG Envelope**

The following XML schema SHOULD be used for any textual instantiation of the IMG envelope:



BEGIN

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:xml:ns:img-envelope"
  elementFormDefault="qualified" attributeFormDefault="unqualified"
  targetNamespace="urn:ietf:params:xml:ns:img-envelope">
  <xs:element name="metadataEnvelope"
    type="metadataEnvelopeType"/>
  <xs:complexType name="metadataEnvelopeType">
    <xs:sequence>
      <xs:element name="item" type="metadataEnvelopeItemType"
        minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="metadataEnvelopeItemType">
    <xs:sequence>
      <xs:element name="metadataFragment" type="xs:string"
        minOccurs="0" maxOccurs="1"/>
      <xs:element name="alternativeURL" type="xs:anyURI"
        minOccurs="0" maxOccurs="unbounded"/>
      <xs:any namespace="##other" minOccurs="0"
        maxOccurs="unbounded" processContents="lax"/>
    </xs:sequence>
    <xs:attribute name="metadataURI" type="xs:anyURI"
      use="required"/>
    <xs:attribute name="version" type="xs:positiveInteger"
      use="required"/>
    <xs:attribute name="validFrom" type="xs:dateTime"
      use="optional"/>
    <xs:attribute name="validUntil" type="xs:dateTime"
      use="optional"/>
    <xs:attribute name="contentType" type="xs:string"
      use="optional"/>
    <xs:anyAttribute processContents="skip"/>
  </xs:complexType>
</xs:schema>
```

END

[Appendix A](#) provides some IMG envelope XML instance examples which use this schema.

This schema enables also use of the index referencing envelope. Thus the element "metadataEnvelope" can consist of multiple "item" elements, which are the actual IMG envelopes. When an envelope object is not an index referencing envelope there SHALL BE only one "item" element.

The element "metadataFragment" contains the embedded IMG metadata





fragment. The IMG metadata fragment MUST NOT be embedded where referencing IMG envelope is used. The index referencing envelope SHALL contain only referencing IMG envelopes.

As stated in the previous section, the contentType attribute is mandatory for any IMG envelope with an IMG metadata fragment embedding within it. The contentType is shown as optional in the above schema as it may be omitted for referencing IMG envelopes.

An embedded IMG metadata fragment SHALL be escaped.

Generally, an embedded IMG metadata fragment SHOULD be escaped by placing inside a CDATA section [17]. Everything starting after "<![CDATA[" string and ending at the "]]>" string would be ignored by the XML parser (quotes not included). Thus, the embedded parts would appear as "<![CDATA[" + IMG\_metadata\_fragment + "]]>". In this case, the complete IMG envelope with embedded IMG metadata fragment MUST NOT violate the rules of CDATA section usage [17].

In the case of an IMG metadata fragment including the XML for a CDATA section, the embedded IMG metadata fragment MAY be escaped by replacing illegal characters with their ampersand-escaped equivalents [17] (instead of encapsulating the whole IMG metadata fragment in a CDATA section). For instance "<" is an illegal character that would be replaced by "&lt;". This method is useful to avoid nesting CDATA sections (which is not allowed).

An IMG metadata fragment which does not adhere to either of these two methods MUST NOT be embedded in an IMG envelope, thus it may only be referenced from an IMG envelope.

Other strategies, such encoding binary IMG metadata fragments as base64 [9], could be useful. However, further specification of the correct structuring IMG metadata fragments to meet character escaping requirements for embedding is beyond the scope of this document.



## **5. Security Considerations**

The IMG envelope is not active content and so MUST NOT be used as executable code. In particular, an IMG envelope MUST NOT be instantiated as a self extracting archive, or indeed in any executable or script form. The XML IMG envelope specified in this document meets these criteria.

Security issues associated with the media types described under the IANA Considerations section of this document have been investigated and no relevant problems have been identified.

## **6. IANA Considerations**

This specification contains two separate items for IANA Considerations:

1. Registration Request for XML Schema of IMG Envelope (urn:ietf:params:xml:schema:img-envelope).
2. Media-Type Registration Request for application/envelope+xml.

### **6.1. Registration Request for XML Schema of IMG Envelope**

Document [18] defines an IANA maintained registry of XML documents used within IETF protocols. The following is the registration request for the IMG Envelope XML schema.

URI:  
urn:ietf:params:xml:schema:img-envelope

Registrant Contact:  
Rod Walsh (rod.walsh (at) nokia.com)

XML: The XML Schema specified in [Section 4.2](#).

### **6.2. Considerations for application/envelope+xml Media-Type Registration**

The specified XML IMG envelope functions as an actual media format of use to the general Internet community and thus media type registration under the Standards Tree is appropriate to maximize interoperability.

One subtype registration, of the application type, is requested:

type-name = application  
subtype-name = envelope+xml

"application/envelope+xml" shall describe the XML IMG envelope format and syntax as specified in [Section 4.2](#) of this document.

#### **6.2.1. Media-Type Registration Request for application/envelope+xml**

This section provides the registration request, as per [5], [6] and [7], to be submitted to IANA following IESG approval.

Type name: application

Subtype name: envelope+xml



Required parameters: none

Optional parameters: none

Encoding considerations: The envelope+xml type consists of UTF-8 ASCII characters [8] and must be well-formed XML.

Additional content and transfer encodings may be used with envelope+xml files, with the appropriate encoding for any specific file being entirely dependant upon the deployed application.

Restrictions on usage: Only for usage with IMG envelopes which are valid according to the XML schema of [Section 4.2](#).

Security considerations: envelope+xml data is passive, and does not generally represent a unique or new security threat. However, there is some risk in sharing any kind of data, in that unintentional information may be exposed, and that risk applies to envelope+xml data as well.

Interoperability considerations: none

Published specification: The present document including Sections [4.1](#) and 4.2.

Applications which use this media type: Not restricted to any particular application.

Additional information:

    Magic number(s): none

    File extension(s): An IMG envelope may use the extension ".env" but this is not required.

    Macintosh File Type Code(s): none

Person & email address to contact for further information: Rod Walsh (rod.walsh (at) nokia.com)

Intended usage: Common

Author/Change controller: Rod Walsh (rod.walsh (at) nokia.com)



## **7. Acknowledgements**

We would like to extend special thanks to Toni Paila, Jean-Pierre Evain, Harsh Mehta and Joerg Ott whose support for, review of, and input to this specification has been very helpful.



## **8. References**

### **8.1. Normative References**

- [1] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [RFC 2119](#), [BCP 14](#), March 1997.
- [2] Nomura, Y., Walsh, R., Luoma, J-P., Ott, J., and H. Schulzrinne, "Requirements for Internet Media Guides (IMGs)", [RFC 4473](#), May 2006.
- [3] Nomura, Y., Walsh, R., Luoma, J-P., Asaeda, H., and H. Schulzrinne, "A Framework for the Usage of Internet Media Guides (IMGs)", [RFC 4435](#), April 2006.
- [4] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- [5] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", [RFC 4288](#), [BCP 13](#), December 2005.
- [6] Freed, N. and J. Klensin, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", [RFC 4289](#), [BCP 13](#), December 2005.
- [7] Murata, M., St.Laurent, S., and D. Kohn, "XML Media Types", [RFC 3023](#), January 2001.
- [8] Yergeau, F., "UTF-8, a transformation format of ISO 10646", [RFC 3629](#), November 2003.

### **8.2. Informative References**

- [9] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", [RFC 3548](#), July 2003.
- [10] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", [RFC 2046](#), November 1996.
- [11] Paila, T., Luby, M., Lehtonen, R., Roca, V., and R. Walsh, "FLUTE - File Delivery over Unidirectional Transport", [RFC 3926](#), October 2004.
- [12] Holbrook, H. and B. Gain, "Source-Specific Multicast for IP", [RFC 4607](#), August 2006.



- [13] Handley, M., Jacobson, V., and C. Perkins, "SDP: Session Description Protocol", [RFC 4566](#), July 2006.
- [14] Deering, S., "Host Extensions for IP Multicasting", [RFC 1112](#), STD 5, August 1989.
- [15] Ramsdell, B., "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification", [RFC 3851](#), July 2004.
- [16] Eastlake, D., Reagle, J., and D. Solo, "(Extensible Markup Language) XML-Signature Syntax and Processing", [RFC 3275](#), March 2002.
- [17] Bray, T., Paoli, J., Sperberg-McQueen, C., Maler, E., Yergeau, F., and J. Cowan, "Extensible Markup Language (XML) 1.1", W3C Recommendation, February 2004.
- [18] Mealling, M., "The IETF XML Registry", [RFC 3688](#), January 2004.



## [Appendix A](#). IMG Envelope Examples

### [A.1](#). SDP Metadata Fragment Embedded in an IMG Envelope

Figure 2. gives an informational example of a Complete IMG Description in SDP [13] embedded within the XML IMG envelope.

```
<?xml version="1.0" encoding="UTF-8"?>
<metadataEnvelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:ns:img-envelope
    ietf-img-envelope.xsd">
  <item
    metadataURI="http://www.example.com/img001/session001.sdp"
    version="1"
    validFrom="2005-12-15T09:30:47-05:00"
    validUntil="2005-12-16T09:30:47-05:00"
    contentType="application/sdp">
    <metadataFragment>
      v=0
      o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
      s=SDP Seminar
      i=A Seminar on the session description protocol
      u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
      e=mjh@isi.edu (Mark Handley)
      c=IN IP4 224.2.17.12/127
      t=2873397496 2873404696
      a=recvonly
      m=audio 49170 RTP/AVP 0
      m=video 51372 RTP/AVP 31
      m=application 32416 udp wb
      a=orient:portrait
    </metadataFragment>
  </item>
</metadataEnvelope>
```

Figure 2: Example of an embedding IMG envelope

### [A.2](#). An IMG Envelope Referencing an XML Metadata Fragment

Figure 3. gives an informational example of the XML IMG envelope referencing an IMG metadata fragment.



```
<?xml version="1.0" encoding="UTF-8"?>
<metadataEnvelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:ns:img-envelope
                      ietf-img-envelope.xsd">
  <item
    metadataURI="http/www.example.com/img001/service001.xml"
    version="1"
    validUntil="2005-12-16T09:30:47-05:00" />
</metadataEnvelope>
```

Figure 3: Example of a Referencing IMG Envelope

### [A.3.](#) An Index Referencing Envelope

Figure 4. gives an informational example of the index referencing envelope referencing multiple IMG metadata fragments.

```
<?xml version="1.0" encoding="UTF-8"?>
<metadataEnvelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:ns:img-envelope
                      ietf-img-envelope.xsd"
  <item
    metadataURI="http/www.example.com/img001/service001.xml"
    version="1"
    validUntil="2005-12-16T09:30:47-05:00" />
  <item
    metadataURI="http/www.example.com/img001/service002.xml"
    version="1"
    validUntil="2005-12-16T09:30:47-05:00" />
  <item
    metadataURI="http/www.example.com/img001/service003.xml"
    version="1"
    validUntil="2005-12-16T09:30:47-05:00" />
  <item>
    metadataURI="http/www.example.com/img001/service004.xml"
    version="1"
    validUntil="2005-12-16T09:30:47-05:00" />
</metadataEnvelope>
```

Figure 4: Example of an Index Referencing Envelope





## [Appendix B](#). IMG Architecture Terminology

NOTE: The following can be replaced by a reference to an eventual IMG architecture specification which includes these definitions.

### Complete IMG Description

Provides a complete syntax and semantics to describe an IMG metadata fragment, which does not need any additional information from other IMG descriptions. It may contain either a full IMG or, more commonly, a subset thereof.

### Delta IMG Description

Provides an update to a Complete IMG Description, defining the difference from the Complete IMG Description in question. This description may be used to reduce network resource usage for instance when small and frequent changes occur to Complete IMG Description. Thus, this description itself cannot represent complete metadata set until it is combined with the corresponding Complete IMG Description.

### Full IMG

Represents a subset/whole of the sender's IMG database delivered within the scope of one IMG transport session. (In this context, the sender's database represents the metadata which the sender has access to, some or all of the IMG metadata may be stored locally on the IMG sender in question.) A sender may deliver only a subset of metadata from its whole metadata database as a full IMG, but a full IMG could also represent the whole IMG database of a particular sender. Different subsets of the sender's IMG database may be provided within different transport sessions; similarly, the same subset may be provided in more than one transport session. Federations of senders using the same definition of full IMG are allowed.

### Internet Media Guide (IMG)

An IMG is a generic term to describe the formation, delivery and use of IMG metadata. The definition of the IMG is intentionally left imprecise.

### IMG ANNOUNCE

IMG ANNOUNCE is an IMG operation for the unsolicited delivery of IMG metadata from an IMG sender to IMG receiver(s).



References to parts of IMG metadata may also be included, instead of the actual metadata.

#### IMG Description

An IMG metadata fragment in one of three forms: Complete IMG Description, Delta IMG Description and IMG Pointer.

#### IMG Element

The smallest atomic element of IMG metadata that can be transmitted separately and referenced individually from other IMG elements.

#### IMG Metadata

A set of metadata consisting of one or more IMG elements. IMG metadata may describe many things, such as the features of multimedia content used to enable selection of and access to media sessions containing content. For example, metadata may consist of the URI, title, airtime, bandwidth needed, file size, text summary, genre, and access restrictions.

#### IMG Metadata Fragment

IMG metadata that is identified distinctly from other IMG metadata. The uniqueness of the identification will be a function of the administrative scope required. An IMG metadata fragment is distinct from an IMG element in that the former may contain multiple IMG elements and the mapping of IMG elements to IMG metadata fragments, for transport, may be deployment specific - even if some of the same content, and IMG elements descriptions of that content, are common to multiple deployments.

#### IMG Metadata Object

A distinct and individually transportable set of IMG metadata such as an IMG metadata fragment or an IMG envelope with an embedded IMG metadata fragment.

#### IMG NOTIFY

Delivery of an IMG update notification in response to an IMG SUBSCRIBE. Identifies the parts of the IMG metadata that have changed without delivering the updated IMG metadata.



## IMG Operation

An atomic process for the IMG transport protocol to deliver IMG metadata or control IMG sender(s) or IMG receiver(s).

## IMG Pointer

Provides a reference that the receiver is able to address specific metadata with. An IMG Pointer may be used to separately obtain (transport) IMG elements, or perform another IMG management function such as data expiry and erasure.

## IMG QUERY

Request to receive a delivery of IMG metadata.

## IMG Receiver

A logical entity that receives media guides from an IMG sender, analogous to a client.

## IMG RESOLVE

Delivery of IMG metadata in response to an IMG QUERY. References to parts of the IMG metadata may also be included, instead of the actual metadata.

## IMG Sender

A logical entity that delivers IMG metadata to one or more IMG receivers, analogous to a server. A sender shall provide bandwidth control or congestion control schemes on the output. A sender can additionally be a receiver - see IMG transceiver.

## IMG SUBSCRIBE

A request for notifications of changes in IMG metadata updates, from a receiver to a sender or transceiver.

## IMG Transceiver

An IMG transceiver combines an IMG receiver and sender. It may modify received IMG metadata or merge IMG metadata received from a several different IMG senders.

## IMG Transport Protocol



A protocol that transports IMG metadata from IMG sender to IMG receiver(s).

#### IMG Transport Session

An association between an IMG sender and one or more IMG receivers within the scope of an IMG transport protocol. An IMG transport session involves a series of IMG transport protocol interactions that provide delivery of IMG metadata from the sender to the receiver(s).

#### IMG Update Notification

Contains IMG Pointer(s) that identify the changed parts of an IMG metadata fragment. An IMG update notification is similar to a Delta IMG Description, with the exception that the changed IMG metadata is not included in this IMG description.





## **Appendix C. IMG Architectural Description**

### **C.1. IMG Metadata Structure and Fragmentation**

#### **C.1.1. Fragmentation**

An IMG sender SHALL select the subset (or entirety) of available IMG metadata that it will make available to IMG receivers using IMG ANNOUNCE and/or IMG RESOLVE. This represents a sender's full IMG and delimits the quantity and level of dynamics a sender must maintain.

The IMG sender SHALL structure its full IMG as a set of IMG metadata fragments, each corresponding to a Complete IMG Description. To IMG transport protocols, an IMG metadata fragment is a discrete unit that can be uniquely identified and versioned.

Note, multiple IMG senders may form a federation of senders/servers and share a common definition of their full IMG and fragmentation structure, and this may additionally be administered by some other entity in the same domain. This document intentionally provides no advice or requirements on federations of senders.

#### **C.1.2. Fragments within Fragments**

An IMG metadata fragment may contain some or all of the description of one or more other IMG metadata fragments. However, this requires a more complex data model, namespace and metadata maintenance mechanisms in both senders and receivers. Some data syntaxes provide tools to simplify the implementation of this kind of feature, such as XPath in XML. However, the use of a general container mechanism must be considered wherever using multiple syntaxes and syntaxes without built-in namespace tools. Grouping mechanisms at the transport layer and Multipart MIME [10] are particularly good candidates for a virtual container and a fully encapsulating container. Any such container is a distinct IMG metadata fragment.

### **C.2. IMG Metadata Discovery**

The following sections describe the use of IMG envelope to support both initial and update discovery of IMG.

#### **C.2.1. Initial IMG Metadata Discovery**

An IMG receiver may need to receive a larger amount of IMG metadata when the terminal has just started receiving from a particular IMG sender, or when its cached copies of IMG metadata cannot be synchronized with IMG updates or have been outdated.



The IMG receiver MUST maintain the version numbers of each IMG metadata fragment to avoid duplication and for update discovery. How the IMG receiver knows when it has received all the IMG metadata fragments it requires (i.e. the determination of an IMG receiver's "full IMG") is out of the scope of this document.

### **C.2.2. IMG Metadata Update Discovery**

Once the IMG receiver has received and stored sufficient IMG metadata in its local database, it may try to detect any changes in the received IMG information. An IMG receiver may monitor the following types of IMG metadata from an IMG sender for changes:

1. Complete description transfers (IMG ANNOUNCE or IMG RESOLVE)
2. Delta description transfers (IMG ANNOUNCE or IMG RESOLVE)
3. IMG update notifications, i.e. Pointer transfers (IMG NOTIFY, IMG ANNOUNCE or IMG RESOLVE)

The receiver will learn of any changes in the IMG metadata by continuing to receive the complete description transfers, for example by periodically using an IMG RESOLVE, or by receiving transmissions of the metadata via IMG ANNOUNCE. However, the use of delta description transfers and/or IMG update notifications may provide more efficient means for update discovery.

A Delta IMG Description provides only part of a Complete IMG Description, defining the difference from a previous version of the Complete IMG Description in question. An IMG receiver MAY ignore delta descriptions and MAY silently discard them: this allows simple "complete-only" receivers to be used wherever the complexity of implementing the delta mechanisms in those receivers is not justified, while not preventing forward migration to delta functionality in the same deployment. (There may be utility in using a only the metadata management (e.g. versioning) information of a delta even in the case of ignoring delta metadata (i.e. using just the pointer data from a delta), however this is beyond the scope of this document.)

The content type of an IMG metadata fragment identifies whether it is a delta description. Other means of delta description type identification, such as dedicated delta transport channels, are also permitted but are beyond the scope of this document. In the absence of another delta description type identification type, the transport mechanism SHOULD identify the content type (i.e. using Content-Type in the HTTP header [\[4\]](#) and Content-Type in the FLUTE FDT Instances



[11])). In the case of an embedding IMG envelope, the content type MUST be given by the IMG envelope, and there is no further requirement on the transport mechanism to declare this.

As with a complete description, a delta description's content type enables a receiver to identify which component should handle this data (i.e. is capable of patching that type of delta description). The content type of the base version (complete description) may also be useful in selecting and configuring the patching component's.

Delta IMG Description content types SHOULD be IANA registered.

The default behaviour of a delta-capable IMG receiver is to interpret the version of the delta description as a single increment from the previous version. In the default case, before applying the incremental patch the receiver MUST have full knowledge of the previous version (it must have received a complete description of the previous version, or have reconstructed it using previous delta description(s) and an earlier version of complete description). Otherwise, the receiver MUST NOT apply the incremental patch. Other patching behaviour, such as multiple minor delta versions to be applied to a common major complete version, are permitted but require additional signalling to the receiver(s) and are beyond the scope of this document.

Where deltas are deployed, an IMG sender SHALL deliver Delta IMG Descriptions using IMG ANNOUNCE, IMG RESOLVE or both operations. After receiving sufficient IMG metadata, an IMG receiver may continue receiving only delta descriptions, if available, instead of complete descriptions. Each delta description describes the IMG metadata (of an IMG metadata fragment) that has recently changed. The definition of 'recently changed metadata' shall be determined by the sender (this may be dependent on time, data size and/or number of transmissions).

After each delta transfer, the IMG receiver MAY need to verify if it has missed an earlier delta transfer(s) to the particular IMG metadata fragment; this can be accomplished by checking the version field in the IMG envelope. The IMG receiver may attempt to recover the missing update by verifying the current versions of the relevant metadata (for example, by obtaining the complete transfer again, or by querying the versions of the locally cached IMG metadata). Note, whether or not a receiver needs to get missing updates is implementation specific.

In addition to sending complete and delta transfers, an IMG sender MAY send IMG update notifications (IMG Pointers). These IMG update notifications consist of references to IMG elements that have changed



recently (e.g. since the previous complete description). After receiving an IMG update notification and discovering the parts of IMG that have changed, an IMG receiver MAY obtain the update from complete or delta descriptions using either IMG ANNOUNCE or IMG QUERY.

### **C.3. Support for Non-contiguous IMG Metadata Fragment Version Series**

The system and transport protocol determines the delivery frequency and delivery methods of IMG envelopes and IMG metadata fragments. However, two specific rules apply in the case that an IMG sender updates an IMG metadata fragment more than once between subsequent deliveries:

- The delta data type MUST NOT be used if the previous version IMG metadata fragment was not delivered to the same receiver-base. Note, "receiver-base" may be a single receiver, for unicast IMG transport protocols, or a user group, for multicast IMG transport protocols. With the possible exception of a case where a receiver confirms multicast delivery to a sender, this implies that delta descriptions must be preceded by zero or more delta descriptions each one version apart and, preceding those, a complete descriptions using the same transport protocol context - i.e. the same protocol and with some context (transaction history) preserved.
- IMG receiver SHALL support reception of subsequent IMG metadata fragment versions, also where the version number has increased by more than one (i.e. where one or more intermediate versions were not send or else lost in delivery).

Note: If the previous received version is earlier than one less than the latest received version, and the latest version delivered is a delta description, the receiver should assume an error has occurred. It may not be possible to determine whether the missed intermediate versions were due to sender, delivery or receiver error. Further discussion of action upon detection of such an error is out of scope of this document.

### **C.4. Detecting the IMG Metadata Fragment Format Type**

For IMG metadata fragments associated with their IMG envelope (not embedded) the MIME type of the IMG metadata fragment SHOULD be provided by the IMG transport protocol (e.g. as a Content-Type text string or a well-specified binary encoded enumeration).

Use of IANA registered MIME types is RECOMMENDED to ensure maximum interoperability. The specific IMG metadata fragment formats (and syntaxes) supported may be implementation depended, although the





following IANA registered MIME types may be of particular interest:

- application/sdp
- application/xml
- multipart/mixed

#### **C.5. Reliable Delivery of IMG Metadata**

Reliable delivery of IMG metadata is a feature of the IMG transport protocol(s) used and the technology suited to providing this very much depends on the receiver base/group size and the selection of unicast/multicast and bi-directional/unidirectional transport. Reliability technology candidates include ACK, NACK, resend, repetition and FEC schemes.

#### **C.6. Parsing and Storage of IMG Metadata Related to Delta Types**

The IMG transport protocol(s) should support all three IMG data types (complete, delta and pointer). (This only implies that the use of any data type shall not break the IMG transport protocol and does not imply that a sender uses all three data types or that a receiver harvests all three.) An IMG transport protocol MUST NOT assume that an IMG metadata fragment is a complete description. An implementation of an IMG transport protocol, IMG envelope management and IMG metadata fragment management may or may not be a single software entity. However, an IMG receiver (software) component that is not capable of correctly using delta descriptions SHOULD NOT process them and it SHOULD hand them to a component that is capable of producing a new IMG metadata fragment by correctly patching the previous version with the delta.

A simple IMG receiver MAY discard delta descriptions in the same way it would discard complete descriptions whose MIME type is unknown. However, in the case where the proportion of such simple IMG receivers in the receiver-base is significant, the system design MAY be, preferably, limited to not send delta descriptions and so avoid the increased diversity in level of sender-receiver data consistency. However, both of these approaches are generally NOT RECOMMENDED.

#### **C.7. Security Considerations**

IMG receivers SHOULD only trust IMG metadata received from a trusted source, with data integrity and authentication of the original IMG sender provided at IMG metadata level or by IMG transport protocol. IMG receivers also SHOULD NOT trust IMG metadata modified by an IMG transceiver, unless the IMG transceiver is trusted and then integrity and authenticity of the changes must be similarly verified. However, to operate in a typical network environment lacking infrastructure



for key distribution and trust verification, IMG receivers MAY be configured to accept untrusted IMG metadata.

There may also be a need to provide access control to the content described by the IMG or to protect the confidentiality of an individual user requesting a particular subset of an IMG. Such privacy requirements SHALL be fulfilled by the use of encryption at IMG metadata level or by IMG transport protocol or IP network protocol.

For multicast delivery of IMG metadata it is also recommended that SSM [[12](#)] rather than ASM [[14](#)] delivery is used so that tampered IMG envelopes can be limited to man-in-the-middle attacks.

The following guidance is provided for designers and implementers of specific IMG metadata fragments. If there is any active content within an IMG metadata fragment, take care to identify, document and (if reasonable) solve the security risks associated with your use of active content. A delta description would normally be used by a "patching application" and so delta description might include data on actions for such an application that could resemble a script (e.g. remove this attribute, change that value). Thus the design and implementation of any "delta patching application" for use with IMGs must address security risks arising from the potential use of a delta description as active content. The authors do not anticipate that Java (or other) application code will be designated as IMG metadata.



Authors' Addresses

Rod Walsh  
Nokia  
P.O. Box 100 (Visiokatu 1)  
Tampere FIN-33721  
Finland

Email: rod.walsh (at) nokia.com

Juha-Pekka Luoma  
Nokia  
P.O. Box 100 (Visiokatu 1)  
Tampere FIN-33721  
Finland

Email: juha-pekka.luoma (at) nokia.com

Jani Peltotalo  
Tampere University of Technology  
P.O. Box 553 (Korkeakoulunkatu 1)  
Tampere FIN-33101  
Finland

Email: jani.peltotalo (at) tut.fi

Sami Peltotalo  
Tampere University of Technology  
P.O. Box 553 (Korkeakoulunkatu 1)  
Tampere FIN-33101  
Finland

Email: sami.peltotalo (at) tut.fi

Janico Greifenberg  
Universitaet Bremen  
MZH 5520  
Bibliothekstr. 1  
Bremen D-28359  
Germany

Email: jgre (at) tzi.de



## Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

